SAM Data Handling Tutorial

Robert Illingworth

29 June 2015

About SAM

- Data handling for the Tevatron Run II experiments
  - goes all the way back to 1997
  - Recently rewritten to provide up-to-date and more user friendly interfaces

- Provides a metadata catalogue
  - What is in the data?
- And a file location catalogue
  - Where is the data?
- And file delivery and tracking
  - Give me the data + what did I do?

Note on examples

- All code was run using sam_web_client v1_9 with SAM_EXPERIMENT=samdev
- Some commands need authentication with a kx509 certificate

- Lines which I typed in are in bold, the response is in normal weight

```
$ samweb -s server-info
SAMWeb API for samdev
Server version: 2.2.1-8-g4823a4a
Cherrypy version: 3.2.4
SQLAlchemy version: 1.0.4
Connected to: postgresql+psycopg2://
samdbs:***@cspgsdev.fnal.gov:5433/samdev
HTTP User-Agent: SAMWebClient/v1_9 (samweb) python/2.6.6
User information:
  Untrusted identity: illingwo@fermicloud058.fnal.gov
  Authenticated username: illingwo
```

Metadata

- Metadata describes the content of a file
- The standard input format for metadata is JSON

```
$ cat test_file_illingwo_1.raw.json
{
  "file_name" : "test_file_illingwo_1.raw",
  "file_size" : 987654321,
  "file_type" : "data",
  "data_tier" : "raw",
  "runs" : [ ["123456", "physics"] ],
  "online.detector" : "neardet"
}
```

Declaring metadata

```
$ samweb -e samdev declare-file test_md.py
$ samweb get-metadata test_file_illingwo_1.raw
      File Name: test_file_illingwo_1.raw
        File Id: 975310
    Create Date: 2015-06-29T18:36:36+00:00
           User: illingwo
      File Type: data
    File Format: unknown
      File Size: 987654321
       Checksum: (none)
 Content Status: good
      Data Tier: raw
online.detector: neardet
           Runs: 123456 (physics)
```

File locations

- Files can have one or more locations

```
$ samweb add-file-location test_file_illingwo_1.raw
enstore:/pnfs/samdev/some/path/to/file
$ samweb locate-file test_file_illingwo_1.raw
enstore:/pnfs/samdev/some/path/to/file

$ samweb add-file-location test_file_illingwo_1.raw
dcache:/pnfs/samdev/persistent/some/path/to/file
$ samweb locate-file test_file_illingwo_1.raw
dcache:/pnfs/samdev/persistent/some/path/to/file
enstore:/pnfs/samdev/some/path/to/file
```

Querying for files

- You can query files by their metadata parameters

```
$ samweb list-files 'data_tier raw and online.detector
neardet and run_number 123456'
test_file_illingwo_1.raw
```

- See “samweb list-files --help-dimensions” for all the query terms

File lineage

- We can also track relationships between files

```
$ cat test_file_illingwo_1_child_1.dat.json
{
  "file_name" : "test_file_illingwo_1_child_1.dat",
  "file_size" : 987654322,
  "file_type" : "data",
  "data_tier" : "reconstructed",
  "application": {
    "family": "reco",
    "name": "dummy",
    "version": "1.0"
   },
  "runs" : [ ["123456", "physics"] ],
  "parents" : ["test_file_illingwo_1.raw"]
}
```

File lineage

```
$ samweb -e samdev declare-file
test_file_illingwo_1_child_1.dat.json
```

```
$ samweb list-files "ischildof: (data_tier raw and
online.detector neardet and run_number 123456) and
version 1.0"
test_file_illingwo_1_child_1.dat
```

```
$ samweb list-files "isparentof: (file_name
test_file_illingwo_1_child_1.dat)"
test_file_illingwo_1.raw
```

Retrieving files

- Simple file access: look up access url for files

```
$ samweb get-file-access-url test_file_illingwo_1.raw
gsiftp://fndca1.fnal.gov:2811/persistent/some/path/to/
file/test_file_illingwo_1.raw
gsiftp://fndca1.fnal.gov:2811/some/path/to/file/
test_file_illingwo_1.raw
```

- Choose a different access method and filter by location

```
$ samweb get-file-access-url test_file_illingwo_1.raw --
schema=xrootd --location=enstore
xrootd://fndca1.fnal.gov:1094/pnfs/fnal.gov/usr/samdev/
some/path/to/file/test_file_illingwo_1.raw
```

Access by SAM project

- A project is a way of pulling files from a dataset to a processing job.
  - Projects only run on defined dataset definitions
  - A single project can have multiple consumer processes
    - Independent processing streams pulling files from the same dataset

- Basic processing workflow
  - Start project
    - Start process
      - Loop: get next file – process – release file
    - Stop process
  - Stop project

Very simple example

- Create a definition – a saved query

```
$ samweb create-definition
illingwo_test_definition_20150629 "create_date >=
'2015-06-29' and create_date < '2015-06-30' and user
illingwo"

Dataset definition 'illingwo_test_definition_20150629'
has been created with id 6624
$ samweb count-definition-files
illingwo_test_definition_20150629
2
```

Running a simple project

- The samweb run-project command runs a simple project that by default just prints out the access URLs (run it with –v if you want to see the http calls it is making)

```
$ samweb run-project --
defname=illingwo_test_definition_20150629
Started project
illingwo_test_definition_20150629_20150629142302
Started consumer processs ID 15223
gsiftp://fndca1.fnal.gov:2812/persistent/some/path/to/
file/test_file_illingwo_1_child_1.dat
gsiftp://fndca1.fnal.gov:2812/persistent/some/path/to/
file/test_file_illingwo_1.raw
Stopped project
illingwo_test_definition_20150629_20150629142302
```

More complex projects

- We have integrated SAM with art, so that the art processing loop gets the next file from SAM, copies it over with ifdh, runs the art processing loop, then releases the file from SAM and asks for the next file
- Allows tracking which files were processed and which were missed, plus some other useful stuff like transfer times

- See http://samweb.fnal.gov:8480/station_monitor/nova/stations/nova/projects/ for nova monitoring with lots of examples